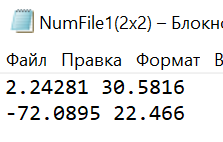
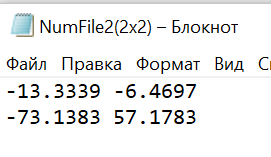
**Отчет о проведенном тестировании**

1. **Проведем тестирование программы на небольших файлах.**
   * **Убедимся в корректности создания и заполнения файлов.**

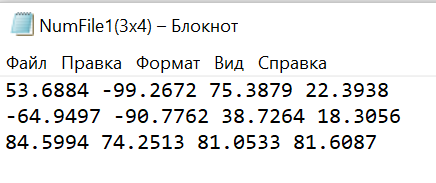
numRows = 2, numCols = 2;



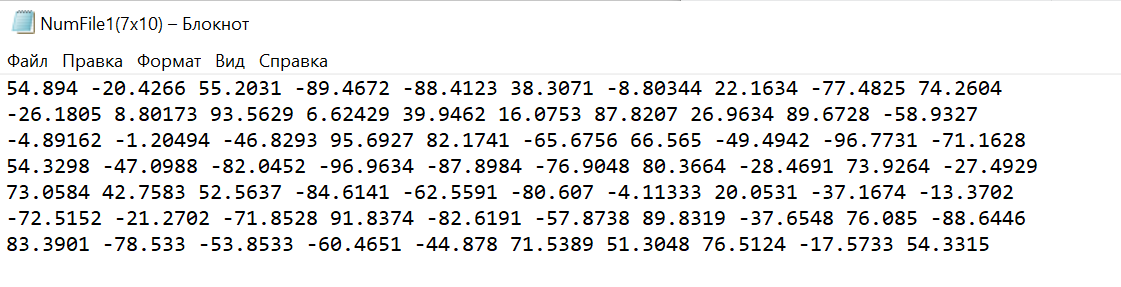
numRows = 2, numCols = 2;



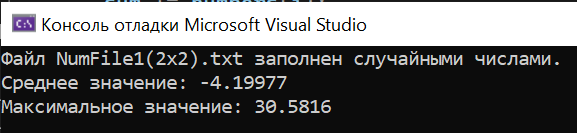
numRows = 3, numCols = 4;



numRows = 7, numCols = 10;

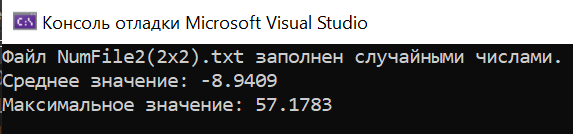


* + **Для первых трех файлов вручную проверим посчитанное значение среднего арифметического и проверим найденное значение максимума.**



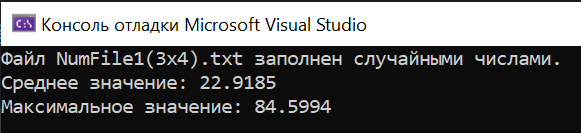
Average = [2.24281 + 30.5816 + (-72.0895) + 22.466] / 4 = -4,19977

Max = 30.5816



Average = [-13.3339 + (-6.4697) + (-73.1383) + 57.1783] / 4 = -8,9409

Max = 57.1783



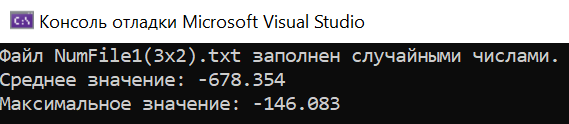
Average = [53.6884 + (-99.2672) + 75.3879 + 22.3938 + (-64.9497) + (-90.7762) + 38.7264 + 18.3056 + 84.5994 + 74.2513 + 81.0533 + 81.6087] / 12 = 22,9185

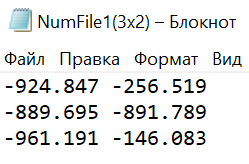
Max = 84.5994

В каждом тесте файл заполнился и сохранился корректно. Значения были найдены верно.

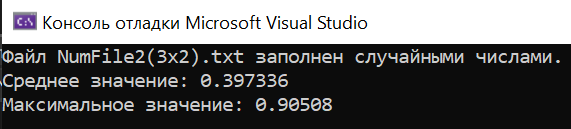
1. **Проверим корректность работы программы при изменении интервала генерации чисел.**

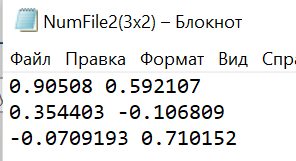
На отрицательном отрезке [-1000; - 100]





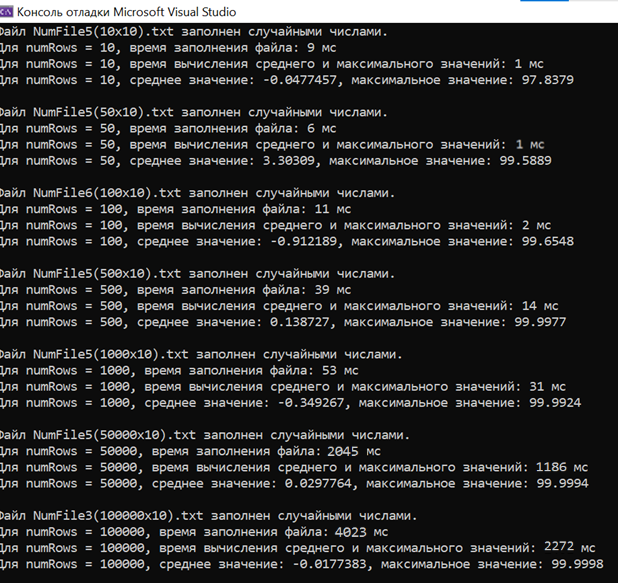
На единичном отрезке [-1; 1]



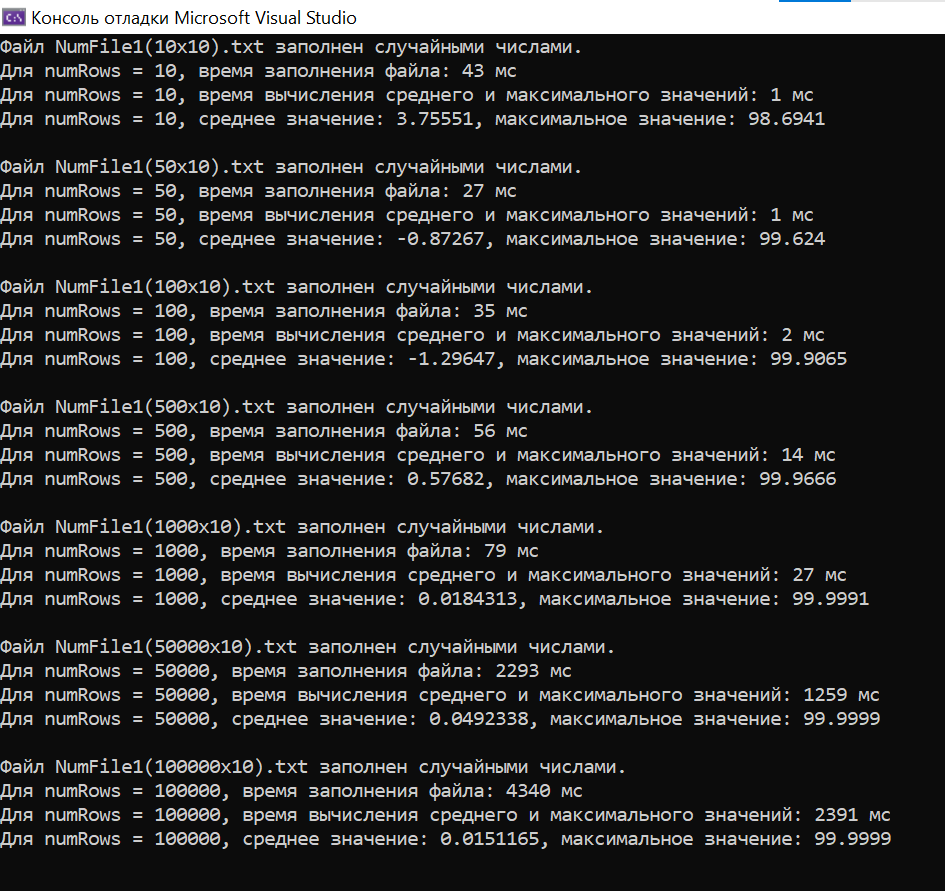


Определение значений корректно.

1. **Добавим счетчики времени в методах с распараллеливанием (заполнение файла случайными числами, поиск максимума и среднего).**
2. **Проверим корректность работы программы при изменении количества строк.**



1. **Уберем распараллеливание для сравнения многопоточной и однопоточной программы.**



**Ускорение многопоточной программы по сравнению с 1 потоком**

Ускорение нелинейно!

****